#5

PATENT

Docket No. JP920000167US1
(590.083)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s)	: Tomio Amano	Group Art: not yet assigned
Serial No.	: not yet assigned	Examiner: not yet assigned
Filed	: herewith	
For	: APPLICATION DATA ERROR CORRECTION SUPPORT	

EXPRESS MAIL CERTIFICATE

Express Mail Label No. EL765475407US

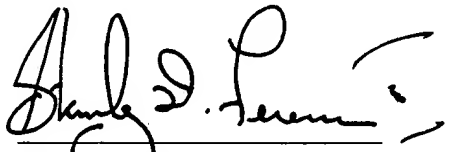
Date of Deposit 27 September 2001

I hereby certify that the following attached paper(s) or fee:

Patent Application
Written Description
Claims 1-22
Abstract
Drawings (Figs. 1-14)
Listing of Inventors
Certified Copy of Priority Application JP2000-295007
Patent Filing Transmittal
Certificate of Express Mail
Two Return Postcards

are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Stanley D. Ference III
(Typed or printed name of person mailing paper)



(Signature of person mailing paper(s) or fee)

Mailing Address:

FERENCE & ASSOCIATES
129 Oakhurst Road
Pittsburgh, Pennsylvania 15215
(412) 781-7386
(412) 781-8390-Facsimile

日　本　国　特　許　庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２０００年　９月２７日

出願番号
Application Number: 特願２０００－２９５００７

出願人
Applicant (s): インターナショナル・ビジネス・マシーンズ・コーポレーション

２００１年　２月２３日

特許庁長官
Commissioner,
Patent Office 及川耕造

出証番号　出証特２００１－３０１１３３２

【書類名】　特許願

【整理番号】　JP9000267

【提出日】　平成12年 9月27日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 11/00

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　天野　富夫

【特許出願人】

【識別番号】　390009531

【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】　100086243

【弁理士】

【氏名又は名称】　坂口　博

【代理人】

【識別番号】　100091568

【弁理士】

【氏名又は名称】　市位　嘉宏

【代理人】

【識別番号】　100106699

【弁理士】

【氏名又は名称】　渡部　弘道

【復代理人】

【識別番号】　100104880

【弁理士】

【氏名又は名称】　古部　次郎

【選任した復代理人】

【識別番号】　100100077

【弁理士】

【氏名又は名称】　大場　充

【手数料の表示】

【予納台帳番号】　081504

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9706050

【包括委任状番号】　9704733

【包括委任状番号】　0004480

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　アプリケーションデータの誤り訂正支援方法、コンピュータ装置、アプリケーションデータ提供システム、および記憶媒体

【特許請求の範囲】

【請求項１】　マークアップを用いた記述用言語にて記述されたアプリケーションデータの誤り訂正支援方法において、

テキストを再入力する際に混入し易い誤りや文字化けを防止するためのタグセットを定義し、

前記記述用言語にて記述される前記アプリケーションデータの所定の部分に対して前記タグセットを用いた書換え情報を付加することを特徴とするアプリケーションデータの誤り訂正支援方法。

【請求項２】　前記タグセットは、同形文字、類似文字、空白、および複雑字形文字の少なくとも何れか１つが存在する文字に対して定義されることを特徴とする請求項１記載の誤り訂正支援方法。

【請求項３】　マークアップを用いた前記記述用言語は、ＸＭＬ(eXtensible Markup Language)であることを特徴とする請求項１記載の誤り訂正支援方法。

【請求項４】　マークアップを用いた記述用言語にて記述されたアプリケーションデータにおける誤り訂正支援方法において、

前記記述用言語で記述される前記アプリケーションデータの要素の中で誤り訂正支援を必要とするテキスト部分を選定し、

選定された前記テキスト部分を所定のタグで囲み、

前記所定のタグで囲まれた前記テキスト部分に対して、所定のアルゴリズムに基づく訂正コードを記述することを特徴とするアプリケーションデータにおける誤り訂正支援方法。

【請求項５】　前記訂正コードは、属性の値および/または属性の名前となる文字列に対して計算され、所定の訂正コード記述用の属性を用いて記述されることを特徴とする請求項４記載の誤り訂正支援方法。

【請求項６】　マークアップを用いた記述用言語にて記述されたアプリケーションデータにおける誤り訂正支援方法において、

前記記述用言語で記述される前記アプリケーションデータの要素の中で誤り訂正支援を必要とする文字列を選定し、

選定された前記文字列に対して所定のアルゴリズムに基づく誤り訂正符号を生成し、

生成された前記誤り訂正符号を前記アプリケーションデータに対する注釈として記述することを特徴とするアプリケーションデータにおける誤り訂正支援方法。

【請求項７】　前記誤り訂正符号は、選定された複数の文字列をまとめて生成され、

生成された前記誤り訂正符号は、前記アプリケーションデータの所定の要素を記述した後に付加されることを特徴とする請求項６記載の誤り訂正支援方法。

【請求項８】　マークアップを用いた記述用言語にて記述されたアプリケーションデータにおける誤り訂正支援方法において、

前記記述用言語で記述される前記アプリケーションデータが有する文脈処理にて支障となる可能性がある単語を所定の属性タイプに分類し、

分類された前記属性タイプを所定のタグセットを用いて前記アプリケーションデータに記述し、

前記属性タイプが記述された前記アプリケーションデータを送出または蓄積することを特徴とするアプリケーションデータにおける誤り訂正支援方法。

【請求項９】　前記所定の属性タイプに分類される文脈処理にて支障となる可能性がある単語は、固有名詞、英語の略称、タグの名前、要素の値として出現するキーワード、属性名、および属性の値として出現するキーワードの少なくとも何れか１つであることを特徴とする請求項８記載の誤り訂正支援方法。

【請求項１０】　マークアップを用いた記述用言語にてアプリケーションデータを生成するコンピュータ装置であって、

前記アプリケーションデータの中における、所定の部分をタグで置き換えるための情報および/または所定の部分に対して誤り検出・訂正コードを計算するための情報が記述されたマークアップ付加用プロファイルと、

前記マークアップ付加用プロファイルを参照して、前記アプリケーションデー

タの所定の部分をタグで置き換えおよび/または当該アプリケーションデータの所定の部分に対して誤り検出・訂正コードを計算し、置き換えられた当該タグおよび/または計算された当該誤り検出・訂正コードを当該アプリケーションデータに付加して訂正情報付きアプリケーションデータを生成するマークアップ付加モジュールと、

前記マークアップ付加モジュールにより生成された前記訂正情報付きアプリケーションデータを出力する出力手段と、を備えたことを特徴とするコンピュータ装置。

【請求項１１】　前記マークアップ付加用プロファイルは、前記誤り検出・訂正コードの情報を前記アプリケーションデータ内に挿入するための情報または前記アプリケーションデータの後ろに注釈として付加するための情報が記述されていることを特徴とする請求項１０記載のコンピュータ装置。

【請求項１２】　マークアップ言語にて生成されたアプリケーションデータを処理可能なコンピュータ装置であって、

所定のテキスト部分がタグで置き換えられる置き換え情報が付加された置き換え情報付きアプリケーションデータを入力する入力手段と、

前記入力手段により入力された前記置き換え情報付きアプリケーションデータにおける前記置き換え情報を認識する認識手段と、

前記認識手段によって認識された前記置き換え情報のタグの表現をテキスト情報に置き換える誤り検出・訂正処理手段と、を備えたことを特徴とするコンピュータ装置。

【請求項１３】　マークアップ言語にて生成されたアプリケーションデータを処理可能なコンピュータ装置であって、

所定のテキスト部分に対して生成された訂正コードが付加された訂正情報付きアプリケーションデータを入力する入力手段と、

前記入力手段により入力された前記訂正情報付きアプリケーションデータにおける前記訂正コードを認識する認識手段と、

前記認識手段によって認識された前記訂正コードを計算して記述されているテキスト部分と比較する誤り検出・訂正処理手段と、を備えたことを特徴とするコ

ンピュータ装置。

【請求項１４】　前記誤り検出・訂正処理手段は、比較の結果、記述されている前記テキスト部分と一致していない場合には、自動訂正可能か否かを判断し、自動訂正が可能である場合には、前記訂正コードに基づく訂正を加えてアプリケーションデータを出力することを特徴とする請求項１３記載のコンピュータ装置。

【請求項１５】　マークアップ言語にて生成されたアプリケーションデータを処理可能なコンピュータ装置であって、

テキスト情報を入力する入力手段と、

入力された前記テキスト情報から認識された個々の文字認識結果と単語辞書とをすり合わせて誤りの検出や修正を行う文脈処理モジュールと、

前記テキスト情報と共に前記入力手段から入力されるタグを利用して前記単語辞書に存在しない単語の情報を認識する単語情報認識手段と、

前記単語情報認識手段により認識された前記単語の情報を前記文脈処理モジュールに提供することを特徴とするコンピュータ装置。

【請求項１６】　マークアップ言語を用いてアプリケーションデータを生成することのできるコンピュータ装置であって、

元となるアプリケーションデータの中から、認識される文字と単語辞書とをすり合わせて誤りの検出や修正を行う文脈処理にて支障となる可能性がある単語を選択する選択手段と、

前記選択手段によって選択された単語に対してタグを用いた誤り訂正コードを記述する記述手段と、

前記記述手段により記述された前記誤り訂正コードを前記アプリケーションデータに付加して出力する出力手段と、を備えたことを特徴とするコンピュータ装置。

【請求項１７】　第１のコンピュータ装置によって生成されたマークアップ言語を用いたアプリケーションデータを第２のコンピュータ装置によって読み込むアプリケーションデータ提供システムであって、

前記第１のコンピュータ装置は、前記第２のコンピュータ装置にてテキストを

再入力する際に混入し易い誤りまたは文字化けを検出するためのタグセットを定義し、定義された当該タグセットを前記アプリケーションデータに付加した訂正情報付きアプリケーションデータを出力し、

前記第２のコンピュータ装置は、前記第１のコンピュータ装置によって出力された前記訂正情報付きアプリケーションデータを入力すると共に、当該訂正情報付きアプリケーションデータに含まれる前記タグセットを認識してアプリケーションデータ中の誤りまたは文字化けを検出または訂正することを特徴とするアプリケーションデータ提供システム。

【請求項１８】　前記第２のコンピュータ装置は、紙ベースの文書または帳票を介して前記第１のコンピュータ装置によって出力された前記訂正情報付きアプリケーションデータを入力することを特徴とする請求項１７記載のアプリケーションデータ提供システム。

【請求項１９】　第１のコンピュータ装置によって生成されたマークアップ言語を用いたアプリケーションデータを第２のコンピュータ装置によって読み込むアプリケーションデータ提供システムであって、

前記第１のコンピュータ装置は、所定のテキストに対して当該テキストに関する付加情報をタグを用いて記述し、記述された当該付加情報を前記アプリケーションデータと共に出力し、

前記第２のコンピュータ装置は、個々の文字認識結果と単語辞書とをすり合わせて誤りの検出や修正を行う文脈処理モジュールを備え、前記第１のコンピュータ装置によって出力された前記アプリケーションデータと前記付加情報とを入力すると共に、入力された前記付加情報を用いて当該文脈処理モジュールにおける当該単語辞書を更新することを特徴とするアプリケーションデータ提供システム。

【請求項２０】　コンピュータに実行させるプログラムを当該コンピュータが読み取り可能に記憶した記憶媒体であって、

前記プログラムは、マークアップ言語にて記述されたアプリケーションデータに含まれるテキストを再入力する際に混入し易い誤りや文字化けを防止するためのタグセットを定義する処理と、当該アプリケーションデータの所定の部分に対

して当該タグセットを用いた書換え情報および/または所定のアルゴリズムに基づく訂正コードを付加する処理と、を前記コンピュータに実行させることを特徴とする記憶媒体。

【請求項２１】　コンピュータに実行させるプログラムを当該コンピュータが読み取り可能に記憶した記憶媒体であって、

前記プログラムは、マークアップ言語にて記述されたアプリケーションデータに含まれるテキスト情報を再入力する際に混入し易い誤りや文字化けを防止するための書換え情報および/または訂正コードが含まれるタグセットを認識する処理と、認識された当該タグセットに基づいて、入力された当該アプリケーションデータにおける所定のテキスト情報を置き換える処理と、を前記コンピュータに実行させること、を特徴とする記憶媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、テキストデータの誤り訂正支援方法等にかかり、特に、紙ベースの文書/帳票と電子化された文書/帳票が混在する環境、あるいはテキスト情報の伝達が確実に行われることが保証できないような環境において、データの交換や蓄積・利用を円滑に行う方法等に関する。

【０００２】

【従来の技術】

電子的に文書を交換するための汎用記述言語として、文書の構造を記述することを重視したマークアップ言語であるＳＧＭＬ(Standard Generalized Markup Language)が存在する。このＳＧＭＬは、文書の論理構造をユーザ自身が定義でき、文書の処理や管理、コンピュータ間におけるデータ交換などが容易に行えることから、文書データを複数のユーザ間で交換する用途に適している。インターネットのWWW(World Wide Web)ページの作成に用いられている記述言語であるＨＴＭＬ(Hyper Text Markup Language)は、このＳＧＭＬを簡略化したものであり、画像や文書を表示するために、＜＞で囲まれたタグと呼ばれる文字列で表示方法を指定することで記述を容易にしている。しかしながら、その一方で、ＳＧＭ

Ｌの有する拡張性が失われている点で問題がある。

【０００３】

一方、電子的な文書/帳票データの交換・蓄積用のフォーマット記述用言語としてＸＭＬ(eXtensible Markup Language)が注目されている。このＸＭＬは、次世代ＨＴＭＬであり、ＳＧＭＬの持つ拡張機能をＷｅｂ上でも利用できるようにした言語仕様である。即ち、文書の構造をＤＴＤ(Document Type Definition：文書型定義)ファイルにすることで、表現方法の指定や文章中の文字列に意味を付加するようなアプリケーション独自のタグを拡張することができる。

【０００４】

このＸＭＬにはいくつか優れた特徴があるが、特に、人が読めるテキストであることと、データとデータを同定するタグによる自己記述的な表現であることが注目に値する。これらの特徴はＸＭＬベースで記述されたデータに対して「フォールバック可能性」と呼ばれる性質をもたらしている。

【０００５】

この「フォールバック可能性」とは、「よい環境でよいアプリケーションを使えば快適ではあるが、貧弱な環境でもそれなりに対処はできる」という性質をいうものと解釈できる。ＸＭＬデータの交換・蓄積では、Ｗｅｂサーバやメールサーバが受信したＸＭＬデータがシームレスにアプリケーションによって処理・格納されるような状況が「よい環境」にあたる。一方、「貧弱な環境」、例えば、自動的なデータ受渡しの機構がない場合でも、人がメールからＸＭＬのタグ付きテキストを切り貼りしてアプリケーションに渡す、受信したＦＡＸの内容(ＸＭＬのタグ付きテキスト)をキー入力してアプリケーションに渡す、といった代替手段をとることができる。バイナリのデータフォーマット、あるいは、ＣＳＶ(Comma Separated Value：データを項目ごとにカンマで区切って羅列するファイル形式)のように、データの値だけが記述されるようなデータフォーマットにおいては、代替手段をとるために追加のツール開発やデータ自体には記述されていない知識(フィールドの順番や位置)が必要となることが多い。

【０００６】

フォールバック可能性を備えたデータ記述を用いるアプリケーションでは、そ

の構成要素となる企業/部門システムやプログラムモジュールに関して様々なレベルでの実現/運用の混在が許容されている。電子的ワークフローに参加したいがＩＴにあまり投資できない企業/部門では、内部の処理や後段への処理済データの受渡しは全て人手で行う場合や、発生頻度の低い要求に関しては人手で対処するといった運用が可能になるのである。マーケットプレイスやサプライチェーンなど規模の異なる独立した企業が参加する(多数が参加するほど価値が高まる)アプリケーションにおいては、このデータ記述のフォールバック可能性の持つ意義は大きい。また、システムをインクリメンタルに開発する、デバッグする等の状況においても有効である。

【０００７】

【発明が解決しようとする課題】

しかしながら、フォールバックをより確実に、より容易に行いたいという観点から見ると、ＸＭＬによるデータ記述にもいくつかの不十分な点がある。その一つは、紙のレベルで代替されたデータ記述の再入力に関する問題である。理屈上では、紙にプリントされたＸＭＬテキストであってもキー入力すれば電子的に作成された元データと同じ内容を再現することができる。しかし、実際には、見た目では解らない空白の数や同じ形の文字/記号があったとき(例えば、マイナスとハイフンなど)、どちらの文字/記号を入力するか、等の問題があり、その結果として微妙に異なるデータが入力されてしまうことがある。人間が読んで内容を理解する場合には問題にならないような差異であるが、例えば、データベースを検索する、署名を検証する、といった処理では不都合が生じてしまう。

【０００８】

また、人手で再入力するのに要する手間も問題である。例えば、ＯＣＲ(Optical Character Reader：光学式文字読み取り装置)のソフトウェアを用いた場合、スキャン解像度等の条件が整えば、９５％から９９％以上の精度でプリントされた文字を読み取ることができる。しかし、残りの１～５％の誤りを確実に見つけるためには、認識されたテキスト全体を人間がチェックしなければならない。認識結果に自信がない部分を警告するＯＣＲは多数、存在しているものの、警告がなされなかった部分が正しく認識されていることを保証しているわけではない。

また、ＯＣＲは、文字ごとの認識結果と単語の辞書とをすり合わせて読み取り精度を高める文脈処理を行っているが、対象テキスト中に辞書にない専門用語やＸＭＬのタグが含まれていると読み取り精度は著しく低下する。再入力の検査と修正に要する人手と時間によっては、ＸＭＬデータの伝達における紙を利用したフォールバックのシナリオが非現実的なものになってしまう。

【０００９】

更に、ＸＭＬのフォールバック可能性を構成する要件として、人間が読んで理解できるテキストベースであることが挙げられるが、テキストでデータ交換を行うが故の問題も発生する。いわゆる文字化けである。例えば、ＸＭＬテキストが幾つかのシステム(サーバ)を経て伝わっていく過程で、非英語圏の文字コードについて異なるエンコーディングを採用しているシステム間での文字コードの変換が行われることがある。変換が常に一意に行われていれば問題はないが、実際にはベンダーごとやバージョンごとに部分的に異なった変換テーブルが使われている。その結果として、例えばＵＴＦ－８ → Shift JIS → ＵＴＦ－８と言う変換を行った際に、一部の文字コードはオリジナルと異なってしまう(化ける)という現象が起こる。ここで、「ＵＴＦ－８」とは、[JIS X 0221]および[Unicode 2.0]の全ての面における文字を表現できる文字符号化スキームである。外字(ISO 10646のプライベート領域に割り当てられた文字)の使用においても同様の問題をもたらす。前の例で、例えば、ＵＴＦ－８で表示・処理を行う者同士では、外字コードに関して合意が成立しているとしても、仲介者がそのコードをShift JISの何というコードに変換するのか、変換されたコードをＵＴＦ－８のどのコードに対応させるのか、といった点が規定されていなければ、外字コードは正しく伝わらない。また更に、インターネット上のデータ交換では、相手や仲介者のシステムの実装を指定することはできない、そもそも知ることができない、という事情もあり、文字化けが発生する危険性が常に存在する。紙からの再入力の場合と同様に、文字化けによるオリジナルとの違いは、例えごく一部であってもデータベース検索や署名検証の処理には致命的な影響を与えてしまう。

【００１０】

また、デジタル・ネットワークを活用したビジネス活動を展開するアプリケー

ションにおいても、ネットワークへの参加を少ない投資で段階的に可能にするという点で、フォールバック可能という性質が持つ意義は大きい。しかしながら、ＸＭＬデータ交換・蓄積において、そのフォールバック可能性をより有効に活用するためには、上記のような問題点を解決する必要がある。

【００１１】

本発明は、以上のような技術的課題を解決するためになされたものであって、その目的とするところは、マークアップによるデータ・文章の記述を行う記述用言語において、テキストを再入力する際に混入し易い誤りや文字化けを防止し、または、これらを検出し、訂正することにある。

また他の目的は、アプリケーションのロジックに依らない汎用的なモジュールとして、記述の付加や誤り検出/訂正を行うプログラムモジュールを提供することにある。

更に他の目的は、最近の技術用語や専門用語、固有名詞等、特別な用語について、ＯＣＲによる文脈処理をフォローできるアプリケーションデータを提供することにある。

【００１２】

【課題を解決するための手段】

かかる目的のもと、本発明は、ＸＭＬ(eXtensible Markup Language)等のマークアップを用いた記述用言語にて記述されたアプリケーションデータの誤り訂正支援方法において、テキストを再入力する際に混入し易い誤りや文字化けを防止するためのタグセットを定義し、このアプリケーションデータの所定の部分に対してタグセットを用いた書換え情報を付加することを特徴としている。

【００１３】

ここで、このタグセットは、同形文字、類似文字、空白、および複雑字形文字(字形が複雑でＦＡＸなどの低解像度のデバイスではイメージが潰れてしまうような文字)の少なくとも何れか１つに対して定義されることを特徴とすれば、例えば紙に印刷したとき、見た目では曖昧性が生じるような文字に対する誤りを軽減できる点で好ましい。

【００１４】

また、本発明の誤り訂正支援方法は、アプリケーションデータの要素の中で誤り訂正支援を必要とするテキスト部分を選定し、選定されたテキスト部分を所定のタグで囲み、所定のタグで囲まれたテキスト部分に対して、所定のアルゴリズムに基づく訂正コードを記述することを特徴としている。

ここで、この訂正コードは、属性の値および/または属性の名前となる文字列に対して計算され、所定の訂正コード記述用の属性を用いて記述されることを特徴とすることができる。

【００１５】

更に、本発明の誤り訂正支援方法は、アプリケーションデータの要素の中で誤り訂正支援を必要とする文字列を選定し、選定された文字列に対して所定のアルゴリズムに基づく誤り訂正符号を生成し、生成された誤り訂正符号をアプリケーションデータに対する注釈として記述することを特徴とすることができる。

【００１６】

ここで、この誤り訂正符号は、選定された複数の文字列をまとめて生成され、生成された誤り訂正符号は、アプリケーションデータの所定の要素を記述した後に付加されることを特徴とすれば、例えば、「以下からは訂正情報である」といったようにまとめて記述することが可能となり、ユーザにとって見易いアプリケーションデータを提供できる点で優れている。

【００１７】

また、本発明の誤り訂正支援方法は、アプリケーションデータが有する文脈処理にて支障となる可能性がある単語、即ち、ＯＣＲ処理における文脈処理を行った際に入っているとうまく機能しないと考えられる単語について、所定の属性タイプに分類し、分類された属性タイプを所定のタグセットを用いてアプリケーションデータに記述し、属性タイプが記述されたアプリケーションデータを送出または蓄積することを特徴としている。この「文脈処理にて支障となる可能性がある単語」とは、固有名詞、英語の略称、タグの名前、要素の値として出現するキーワード、属性名、および属性の値として出現するキーワード等の少なくとも何れか１つである。

【００１８】

一方、本発明は、マークアップを用いた記述用言語にてアプリケーションデータを生成するコンピュータ装置であって、アプリケーションデータの中における、所定の部分をタグで置き換えるための情報および/または所定の部分に対して誤り検出・訂正コードを計算するための情報が記述されたマークアップ付加用プロファイルと、このマークアップ付加用プロファイルを参照して、アプリケーションデータの所定の部分をタグで置き換えおよび/またはアプリケーションデータの所定の部分に対して誤り検出・訂正コードを計算し、置き換えられたタグおよび/または計算された誤り検出・訂正コードをアプリケーションデータに付加して訂正情報付きアプリケーションデータを生成するマークアップ付加モジュールと、このマークアップ付加モジュールにより生成された訂正情報付きアプリケーションデータを出力する出力手段とを備えたことを特徴としている。

【００１９】

ここで、このマークアップ付加用プロファイルは、誤り検出・訂正コードの情報をアプリケーションデータ内に挿入するための情報またはアプリケーションデータの後ろに注釈として付加するための情報が記述されていることを特徴とすることができる。

【００２０】

他の観点から把えると、本発明が適用されるコンピュータ装置は、所定のテキスト部分がタグで置き換えられる置き換え情報が付加された置き換え情報付きアプリケーションデータを入力する入力手段と、この入力手段により入力された置き換え情報付きアプリケーションデータにおける置き換え情報を認識する認識手段と、この認識手段によって認識された置き換え情報のタグの表現をテキスト情報に置き換える誤り検出・訂正処理手段とを備えたことを特徴としている。

【００２１】

また、本発明が適用されるコンピュータ装置は、所定のテキスト部分に対して生成された訂正コードが付加された訂正情報付きアプリケーションデータを入力する入力手段と、この入力手段により入力された訂正情報付きアプリケーションデータにおける訂正コードを認識する認識手段と、この認識手段によって認識された訂正コードを計算して記述されているテキスト部分と比較する誤り検出・訂

正処理手段とを備え、この誤り検出・訂正処理手段は、比較の結果、記述されているテキスト部分と一致していない場合には、自動訂正可能か否かを判断し、自動訂正が可能である場合には、訂正コードに基づく訂正を加えてアプリケーションデータを出力することを特徴としている。

【００２２】

更に、本発明が適用されるコンピュータ装置は、例えば紙ベースの文書や帳票からテキスト情報を入力する入力手段と、入力されたテキスト情報から認識された個々の文字認識結果と単語辞書とをすり合わせて誤りの検出や修正を行う文脈処理モジュールと、テキスト情報と共に入力されるタグを利用して単語辞書に存在しない専門用語やＸＭＬタグ等の単語の情報を認識する単語情報認識手段とを備え、認識された単語の情報を文脈処理モジュールに提供して、例えばＯＣＲにおける読み取り精度を向上させることを特徴としている。

【００２３】

また、本発明が適用されるコンピュータ装置は、他のコンピュータ装置にて読み取られる際に、元となるアプリケーションデータの中から、認識される文字と単語辞書とをすり合わせて誤りの検出や修正を行う文脈処理にて支障となる可能性がある単語を選択する選択手段と、この選択手段によって選択された単語に対してタグを用いた誤り訂正コードを記述する記述手段と、この記述手段により記述された誤り訂正コードをアプリケーションデータに付加して、紙等に出力する出力手段とを備えたことを特徴としている。

【００２４】

一方、本発明は、第１のコンピュータ装置によって生成されたマークアップ言語を用いたアプリケーションデータを第２のコンピュータ装置によって読み込むアプリケーションデータ提供システムであって、この第１のコンピュータ装置は、第２のコンピュータ装置にてテキストを再入力する際に混入し易い誤りまたは文字化けを検出するためのタグセットを定義し、定義されたこのタグセットをアプリケーションデータに付加した訂正情報付きアプリケーションデータを出力し、第２のコンピュータ装置は、出力されたこの訂正情報付きアプリケーションデータを入力すると共に、訂正情報付きアプリケーションデータに含まれるタグセ

ットを認識してアプリケーションデータ中の誤りまたは文字化けを検出または訂正することを特徴としている。尚、第２のコンピュータ装置への出力は、紙ベースの文書/帳票の他、電子化された文書/帳票が混在する環境、あるいは、テキスト情報の伝達が確実に行われることが保証できないような環境からなされる場合がある。

【００２５】

また、本発明が適用されたアプリケーションデータ提供システムにて、第１のコンピュータ装置は、所定のテキストに対してテキストに関する付加情報をタグを用いて記述し、記述された付加情報を前記アプリケーションデータと共に出力し、第２のコンピュータ装置は、個々の文字認識結果と単語辞書とをすり合わせて誤りの検出や修正を行う文脈処理モジュールを備え、第１のコンピュータ装置によって出力されたアプリケーションデータと付加情報とを紙ベースの文書または帳票を介して入力すると共に、入力された付加情報を用いて文脈処理モジュールにおける単語辞書を更新することを特徴としている。

【００２６】

更に、本発明は、コンピュータに実行させるプログラムをコンピュータが読み取り可能に記憶した記憶媒体であって、このプログラムは、ＸＭＬ等のマークアップ言語にて記述されたアプリケーションデータに含まれるテキストを再入力する際に混入し易い誤りや文字化けを防止するためのタグセットを定義する処理と、アプリケーションデータの所定の部分に対してタグセットを用いた書換え情報および/または所定のアルゴリズムに基づく訂正コードを付加する処理とをコンピュータに実行させることを特徴としている。

【００２７】

他の観点から把えると、本発明は、コンピュータに実行させるプログラムをコンピュータが読み取り可能に記憶した記憶媒体であって、このプログラムは、マークアップ言語にて記述されたアプリケーションデータに含まれるテキスト情報を再入力する際に混入し易い誤りや文字化けを防止するための書換え情報および/または訂正コードが含まれるタグセットを認識する処理と、認識されたタグセットに基づいて、入力されたアプリケーションデータにおける所定のテキスト情

報を置き換える処理とを前記コンピュータに実行させることを特徴としている。これらの記憶媒体としては、例えばＣＤ－ＲＯＭ媒体等が該当し、コンピュータ装置におけるＣＤ－ＲＯＭ読み取り装置によってプログラムが読み取られ、例えば、コンピュータ装置におけるハードディスクにこのプログラムが格納され、実行される形態が考えられる。

【００２８】

【発明の実施の形態】

以下、添付図面に示す実施の形態に基づいてこの発明を詳細に説明する。

まず最初に、本実施の形態における誤り訂正方法の理解を容易にするために、本実施の形態における誤りの防止・検出・訂正用のマークアップの例について説明する。ここでは、（１）対象データの置き換え、（２）対象データに誤り検出/訂正情報を挿入/追加、（３）対象データの内容に関する情報を追加、の３つの例を挙げて説明する。

【００２９】

（１）対象データの置き換え

紙に印刷したとき見た目では曖昧性が生じるような文字を、特定の要素で置き換えるものである。対象となるのは、空白や同形文字や類似文字が存在する文字、字形が複雑で、ＦＡＸなどの低解像度のデバイスではイメージが潰れてしまうような文字である。

図１は、本実施の形態における対象データの置き換え例を示した図である。ここでは、例えば、半角空白を<ec:sp/>に、全角空白を<ec:sp2/>または<ec:ch utf="x0030"> </ec:ch>に、また、同様にして、「－（マイナス）」、「ー（長音）」、「力（漢字）」、「カ（カタカナ）」を所定の文字コードの記述によって置き換えている。

【００３０】

ここでは、主に、人が紙になったものを入力し直す必要が生じた場合や、ＯＣＲで読み直す必要が生じた場合を想定している。紙になってしまうと、例えば半角の空白が２つであるのか、全角の空白なのか、などは全く理解できないし、見かけ上、同じ形をした文字も存在している。また、複雑な字形で、複写を施した

際に潰れてしまい、ＯＣＲでは読めない、という文字も存在する。本実施の形態では、そういう文字を文字コードの記述によって置き換えることで、その表現は冗長となる場合があるものの、形が似ている文字であっても全く異なるものとして、異なるコードによって読み取ることが可能となる。即ち、本実施の形態における対照データの置き換えでは、英数字を用いて所定のコードを置き換えることで、例えばＯＣＲで読ませる場合であっても、漢字などで読ませる場合に比べて、読み取り率を各段に向上させることができる。

【００３１】

（２）対象データに誤り検出/訂正情報を挿入/追加

まず、要素内のテキストに関する誤り訂正情報を挿入する例について説明する。

図２は、誤り訂正符号の作成例を示した図であり、ここでは、「コンピュータによる帳票処理は」という文字列に対して作成される訂正コード例を示している。本実施の形態では、要素内のテキスト部分を、本実施の形態のために用意したタグで囲み、誤り訂正コードを記述している。この誤り訂正コードの生成には、既存のアルゴリズムを用いることができる。例えば、図２にあるように、１文字１６ビット(例えば、ＵＴＦ-１６：[JIS X 0221]および[Unicode 2.0]の最初の１７面にある全ての文字を表現できる文字符号化スキーム)で表現された文字列に対して、各桁ごとのビット列を想定し、それに対する訂正符号を計算する。例えば、図２に示す各文字の１ビット列(例えば丸で囲まれたビット)に対して、所定のアルゴリズムを適用して所定の計算を行い、「２Ａ」という値を得る。ハミング符号(２つの２進数の間で異なる桁の数を一定以上となるように検査ビットを付け、間違いを訂正できるようにしたもの)を用いれば、訂正コードを各８ビット(１６進２桁)として３２文字分の訂正コードを用意することにより、最大２４７文字の列に対して１文字の認識誤りを訂正することができる。

【００３２】

図３(ａ),(ｂ)は、上述した要素内のテキストに関して誤り訂正情報を挿入した例を説明するための図であり、図３(ａ)は挿入前を、図３(ｂ)は挿入後を示している。ここでは、「ＩＢＭ製パーソナルコンピュータ」という文字列に対して

、属性val_ecの値、文字列に対して計算された訂正コードである「８Ｂ１２……７Ｂ２９」という値が、文字列に付加されている。「ＩＢＭ製パーソナルコンピュータ」という文字列を入力し直したときに、同じようにコード列に対して、同じアルゴリズムを用いて計算を行う。全く誤りがなく入力し直された場合には、図３（ｂ）に示される訂正コードと同一の値が得られるが、どこかに誤りがある場合には、別の値が出力される。計算に用いられるアルゴリズムは、偶然、一致する場合が最も低くなるアルゴリズムが採用されている。入力し直したときに誤りがあった場合には、訂正コードに対する“バケ”ができるので、統計的に高い確率、即ち、ＯＣＲでの読み取り率とは比べものにならない程度の高い確率にて、誤りを認識することができる。

【００３３】

次に、属性の値や名前に関する誤り訂正情報を挿入する例について説明する。

図４（ａ）～（ｃ）は、本実施の形態における訂正コード記述用属性を用いた訂正情報の挿入例を示す図であり、図４（ａ）は訂正コード記述用属性の例を示し、図４（ｂ）はその挿入前を、図４（ｃ）はその挿入後を示している。

ここでは、属性の名前、値または両方の文字列に対して誤り訂正コードを計算し、本実施の形態のために用意した属性の値として記述する。訂正コード生成の対象となる文字列の種類と、訂正コード記述用に本実施の形態で用意した属性との関係は、図４（ａ）に示すようになる。例えば、訂正コード記述用属性である「val_ec」は、「属性の値となる文字列に対する訂正コード」を示し、「name_ec」は「属性の名前となる文字列に対する訂正コード」を、「both_ec」は「属性の名前と値を連結した文字列に対する訂正コード」を示している。

【００３４】

対象となる属性が複数ある場合には、文字列を(例えば属性名のアルファベット順で)連結した文字列に対して、誤り訂正コードを計算する。図４（ｂ）および（ｃ）に示される例では、「ＩＢＭ５５５０」という文字列に対して、誤り訂正コードが計算され、「val_ec」を用いて示されている。訂正コード記述用の属性として「both_ec」を用いた場合には「ccodeＩＢＭpcode５５５０」という文字列に対して誤り訂正コードが計算される。即ち、名前の部分と値の部分とで、間違

いはどちらにも起こり得ることから、名前と値のペアで記述することには意味がある。

【００３５】

これらの例において、長い文字列に対して訂正情報を挿入した場合には、誤り訂正に要するデータ量は少ないが、誤りの箇所が解らなくなる可能性がある。一方、短い文字列に対して訂正情報を挿入した場合には、誤りがどこかを発見し易くなる一方で、データ量が多くなる欠点がある。従って、これらを比較衡量して、選定する文字列の長さが決定される。例えば、属性情報に関しては、あまり文字数がないことから、まとめて誤り訂正コードを計算することが好ましい。

【００３６】

次に、複数の要素や属性の値に関する誤り訂正情報をまとめて記述する例について説明する。

図５(ａ),(ｂ)は、アプリケーションデータの記述の後に誤り訂正情報を付加した例を示した図である。図５(ａ)は、所定の文字列に対する誤り訂正符号を注釈で付けた例を示し、図５(ｂ)は、更に、その誤り訂正符号に対して誤り訂正符号を付加した例を示している。

前述した図３および図４の記述では、アプリケーションデータを表すタグ付きテキスト中に混在する形で誤り訂正情報を記述している。しかしながら、例えば、XPath等を用いてアプリケーションデータに対する注釈のような形で誤り訂正情報を記述することも可能である。例えば図３および図４で使われていた<ProductDescription>要素と<ProductCode>要素によるアプリケーションデータの記述の後に、図５(ａ)に示すように誤り訂正情報を付加することが可能である。即ち、ここで計算されている誤り訂正符号は、文字列「ＩＢＭ製パーソナルコンピュータ５５５０ＩＢＭ」に対するものになる。このように記述することで、アプリケーションはまとめて書いておきたいという要望が強い場合に、以下からは訂正情報であることを明記して誤り訂正情報を付加することが可能となる。

【００３７】

一方、図５(ｂ)に示すように、図５(ａ)に示す記述に対して、更に、誤り訂正情報を付加することもできる。図５(ｂ)の例では、図５(ａ)に示す記述中のval_

ec属性とpath属性の値を、出現順に連結した文字列に対して誤り訂正符合を付加している。このように、ＸＭＬのマークアップを用いることで、必要に応じ、同じようにして誤り訂正符合を付加することが可能となる。

【００３８】

（３）対象データの内容に関する情報を追加

ＯＣＲを用いてテキスト入力する場合、個々の文字認識結果と単語辞書とをすり合わせて誤りの検出や修正を自動的に行う「文脈処理」と呼ばれる処理が有効である。この「文脈処理」とは、個々の文字認識結果と単語辞書とをすり合わせて読み取り精度を高める処理であり、即ち、一つ一つの文字の認識結果と単語辞書との組み合わせによって認識率を良くすることが可能である。しかしながら、この「文脈処理」は、ＯＣＲの辞書にない固有名詞や専門用語、ＸＭＬのタグなどが対象テキスト中に含まれていると、良好に機能しない。ここでは、後述するようなタグを利用して、辞書にない単語の情報を記述し、文脈処理モジュールに与えている。

【００３９】

図６（ａ）～（ｃ）は、ＯＣＲの文脈処理モジュールに対して提供する情報の例を示した図である。図６（ａ）は文脈処理モジュールに対して与えるタグの例を示し、図６（ｂ）は属性タイプの意味を示し、図６（ｃ）は上述の（１）および（２）の手法を更に適用して情報が追加された例を示している。図６（ｂ）に示されるように、タイプ(type)の値「ProperNoun」は「固有名詞」の意味、「Abbreviation」は「英語の略語」の意味等、単語の意味を付加情報として加えている。図６（ａ）の例では、「鈴木一郎」について、タグを利用して「固有名詞」であることを明記し、「ＸＭＬ」には「英語の略語」であることが示され、「ProductCode」には「タグの名前」であること、「ccode」には「属性名」であることが示されている。

【００４０】

このように、一般に、最近の技術用語のごとく頻繁に新しい単語が出現する場合には、ＯＣＲだけで対応することが困難となるが、本実施の形態では、例えば、新しい単語や特別な用語に対して、ＸＭＬのタグの形で文章に付加することで

、それを読み取ったＯＣＲは、その情報を用いて文字認識に役立てることができる。即ち、文脈処理モジュールにて、これらの情報を、その文章に対して認識率を高めるために用いるだけではなく、他の文章に対する認識率の向上に役立てることが可能となる。

【００４１】

尚、これらの記述は、アプリケーションデータに付加され印刷された紙からＯＣＲによって入力されても良いし、別途、電子的データとして送られるか入力を担当する者が手入力しても構わない。紙に印刷される場合には、図６（ｃ）に示すように、上述の（１）および（２）の手法を適用して、誤りやすい文字を置き換えたり誤り訂正情報を付加することができる。ここでは、「固有名詞」であることを明記して、「鈴木一郎」に対する訂正コードと共に、誤り易い「一」は漢字であることを明記している。

【００４２】

以上のようにして、問題解決のために追加した記述が、ＯＣＲによる読み取りや伝達の過程で誤って再入力される可能性もある。しかしながら、以下（①～④）に述べるような理由により、アプリケーションデータ記述部分で誤りが起こる可能性よりも十分に低いと考えられる。

① 上記（１）（２）（３）の記述中に使われる文字種は英数字と一部の記号に限定され、かつ、要素名や属性名、属性値について記述される可能性のある文字列が事前に解っており、文脈処理による精度の向上が期待できること。

② 上記（１）（２）（３）の記述中に、文字化けの可能性のある全角記号等は出現しないこと。

③ 一般にアプリケーションデータ記述よりも文字数が少ないため、文字列全体として正しく認識される可能性が高いこと。

④ 誤り訂正情報の記述に対して更に誤り訂正情報を付加することが可能であること。

【００４３】

次に、上述した方法を実現するために、本実施の形態が適用されたシステムの具体的構成を説明する。

図７は、本実施の形態が適用された誤り訂正支援システムの全体構成を示す説明図である。この例では、第１のコンピュータ装置１０の第１アプリケーション１１と第２のコンピュータ装置２０の第２アプリケーション２１との間、即ち、別々の環境にて動いている第１アプリケーション１１から第２アプリケーション２１に対して、ＸＭＬアプリケーションデータ４０が伝達される。

【００４４】

第１のコンピュータ装置１０は、マークアップ付加用プロファイル１２と、このマークアップ付加用プロファイル１２を参照しながら処理を行う誤り防止・検出・訂正マークアップ付加モジュール１３、また、データ送り出し機構３１を備える場合がある。一方、第２のコンピュータ装置２０は、マークアップ認識用プロファイル２２と、このマークアップ認識用プロファイル２２を参照して処理する誤り検出・訂正モジュール２３とを備え、第２アプリケーション２１を出力している。また、データ受け取り機構３２を備える場合がある。このデータ送り出し機構３１およびデータ受け取り機構３２は、他のモジュールによる構成であっても構わない。

【００４５】

データ伝達部３０は、例えば、第１のコンピュータ装置１０のデータ送り出し機構３１と第２のコンピュータ装置２０のデータ受け取り機構３２により、ネットワーク３３を介してデータを伝達する。また、第１のコンピュータ装置１０側のプリンタ３４によって出力された紙データを人や郵送等により伝達し、第２のコンピュータ２０側のスキャナ＆ＯＣＲ３５によって読み取る場合もある。また、第１のコンピュータ装置１０側でプリントアウトした後にＦＡＸスキャナ３６で読み取られ、電話回線を介してＦＡＸプリンタ３７で出力される場合もある。勿論、第１のコンピュータ装置１０側および/または第２のコンピュータ装置２０側にてプリントアウトされないＦＡＸ送受信の場合もある。このように、データ伝達部３０の部分は、自動的にアプリケーションとトランスポート層を結び付けるＢ２Ｂ(企業対企業)サーバかもしれないし、人がカット＆ペーストで(あるいはＯＣＲを使って)作業している場合もある。また、インターネット上であっても、色々なシステムの間を渡ってデータが伝達された場合に、例えば、コード

体系等が異なるシステムでやり取りがなされる可能性がある。従って、このデータ伝達部３０の部分は、何があるかが解らない部分、即ち、様々なフォールバックシナリオが存在し得る部分として把えることができる。

【００４６】

マークアップ付加用プロファイル１２には、アプリケーションデータ中のどの文字をタグで置き換えるか、どの部分に対して誤り検出・訂正コードを計算するか、訂正コードの情報をアプリケーションデータ内に挿入するかXPathを使ってデータの後ろに付加するか等が記述されており、誤り防止・検出・訂正マークアップ付加モジュール１３はマークアップ付加用プロファイル１２を参照しながら処理を行う。この処理によって、ＸＭＬアプリケーションデータ４０は、一部改変されて書換えＸＭＬアプリケーションデータ４２となり、また、いくらかの誤り防止・検出・訂正用記述４３が追加されて、訂正情報付きアプリケーションデータ４１が生成される。

【００４７】

第１のコンピュータ装置１０側の第１アプリケーション１１が生成した訂正情報付きアプリケーションデータ４１(書換えＸＭＬアプリケーションデータ４２および誤り防止・検出・訂正用記述４３)は、データ伝達部３０により第２のコンピュータ装置２０側に伝達される。即ち、前述したように、例えば、データ送り出し機構３１によりネットワーク３３(ＨＴＴＰやＳＭＴＰなど)、ＦＡＸスキャナ３６、郵送などの伝達手段に渡された後、例えば、データ受け取り機構３２を経て第２アプリケーション２１に受信される。

【００４８】

データを受け取る側として第２のコンピュータ装置２０側における第２アプリケーション２１とデータ受け取り機構３２の間には、誤り検出・訂正モジュール２３が存在しており、マークアップ認識用プロファイル２２に基づいて訂正情報付きアプリケーションデータ４１を解析し、誤りの検出、訂正(必要なら人間による訂正を促す)を行う。訂正処理が全て終了後、誤り検出・訂正モジュール２３は検出・訂正用のタグや属性を削除し、タグを直して、例えばスペース等を形成して、ＸＭＬアプリケーションデータ４０を復元している。

【００４９】

図８は、第１のコンピュータ装置１０側の誤り防止・検出・訂正マークアップ付加モジュール１３における処理を示したフローチャートである。誤り防止・検出・訂正マークアップ付加モジュール１３は、まず、ＸＭＬアプリケーションデータ４０を読み込んで(ステップ１０１)、例えば、ＤＯＭ(Document Object Model)のような内部データ形式に展開する。そして、要素内のテキストに関する誤り訂正情報を挿入し(ステップ１０２)、属性の名前や値を示す文字列に関する誤り訂正情報を挿入する(ステップ１０３)。また、Xpath指定による誤り訂正情報を付加し(ステップ１０４)、対象データの内容に関する情報を追加し(ステップ１０５)、間違え易い文字や空白の置き換えを行う(ステップ１０６)。これらの訂正情報を付加する処理を行った後に、訂正情報付きアプリケーションデータ４１を出力する(ステップ１０７)。本実施の形態ではＸＭＬデータを整形式として扱っている。

【００５０】

図９は、第２のコンピュータ装置２０における誤り検出・訂正モジュール２３内の処理を示したフローチャートであり、ＯＣＲを用いて紙から再入力を行う場合の処理を例として示している。誤り検出・訂正モジュール２３では、まず、ＯＣＲまたは人が入力したテキストファイルからＯＣＲ処理の中間結果を読み込む(ステップ２０１)。この中間結果とはＯＣＲで認識したテキストに２位以下の認識候補の情報を付加したものをいう。

図１０は、この中間結果をＸＭＬベースで記述した例である。ここでは、「これは認識結果です。」という文字列の「こ」と「果」について、２位、３位の候補の情報が付加されている。人が入力したテキストは、１位候補だけで構成された中間結果とみなすことができる。人が入力したテキストに対して、この文字はこちらの文字と間違え易い、という情報が既知であれば、その情報に基づいて２位、３位候補の情報を付加するように構成することもできる。

【００５１】

図９のフローチャートに戻ると、ステップ２０１の後、読み込まれた中間結果に対して、ミニマム単語セットによる文脈処理が行われる(ステップ２０２)。文

脈処理は、ＯＣＲ中間結果のテキストを基本的な語句/単語に分割し、それぞれの単語が辞書に登録されているかチェックする。登録されていない場合、１位候補の文字を２位以下の候補文字と置き換えることにより、登録されている単語に合致させることができるか否かを判定し、可能であれば１位候補文字の入れ替えを行う。文脈処理については、既にアルゴリズムが確立しているので、具体的な実装に関しては既存のものを用いることができる。単語辞書には、一般的な日本語の単語に、上述した方法(１)～(３)にて本実施の形態のために定義されたタグの情報を加えたもの(ミニマム単語セット)を用いる。

【００５２】

次に、対象データの内容に関する情報を記述したテキスト断片の切出しが行われる(ステップ２０３)。即ち、最初の文脈処理が行われた後のテキストから、上述した方法(３)の<word>タグを用いた記述と、それに続く誤り訂正コードの記述が抜き出される。その後、抜き出されたテキストに対して、誤り検出・訂正情報付きテキストの処理が行われる(ステップ２０４)。この処理結果から、固有名詞やアプリケーション固有のタグ情報を抜き出し、文脈処理用の単語辞書に追加することで、単語セットが拡張される(ステップ２０５)。ここで、アプリケーションデータに関するＤＴＤやスキーマが与えられている場合には、それらからタグ名、属性名や、値として出現し得る文字列などの情報を抜き出して、辞書に追加することも可能である。その後、単語を追加した辞書(拡張単語セット)を用いて、再度、文脈処理が行われる(ステップ２０６)。その後、テキスト全体に対して誤り検出・訂正情報付きテキストの処理が行われ(ステップ２０７)、誤り検出・訂正モジュール２３での一連の処理が終了する。尚、一般の文書の入力支援に用いる場合には、ステップ２０１、２０５および２０６によって処理が構成される。また、文字化けに対処する場合には、ステップ２０１からステップ２０６は省略することが可能である。

【００５３】

図１１は、図９のステップ２０４およびステップ２０７で行われる誤り検出・訂正情報付きテキストの処理の概要を示したフローチャートである。まず、ＸＭＬデータの読み込みが行われ(ステップ３０１)、ＸＭＬテキストは、ＤＯＭ(Doc

ument Object Model)のような内部データ形式に展開される。この時点で整形式のＸＭＬテキストでなかった場合には、エラーメッセージに基づいて人間による修正が行われる。次に、上述した方法(１)にて記述されているような、タグによる文字や空白の表現を置き換え、元に戻す処理が行われる(ステップ３０２)。その後、全ての検出訂正情報をチェックしたか否かの判断がなされる(ステップ３０３)。チェックしていない場合には、上述した方法(２)にて記述されているような誤り訂正コードの記述それぞれについて、アプリケーションデータから訂正コードが計算される(ステップ３０４)。そして、計算されたものと記述されている値とが一致しているか否かが判断され(ステップ３０５)、一致している場合には、ステップ３０３の判断に戻る。

【００５４】

一方、ステップ３０５にて、記述されている値と一致していない場合には、自動訂正可能か否かが判断される(ステップ３０６)。自動訂正可能である場合には、訂正コードに基づく訂正が行われ(ステップ３０７)、また、自動訂正が可能でない場合には、人間による訂正が行われ(ステップ３０８)、それらの訂正後に、ステップ３０３の判断に戻る。これらの作業が繰り返され、ステップ３０３にて全ての検出訂正情報のチェックが終了したと判断される場合には、最後に、誤り検出・訂正用のタグや属性が削除されて(ステップ３０９)、オリジナルのＸＭＬアプリケーションデータ４０が出力される(ステップ３１０)。

【００５５】

次に、本実施の形態を用いた４つの応用例について、説明する。

応用例 １）小規模企業や個人利用者による署名つきデータの紙による保存

例えば、Ｂ２ＢやＢ２Ｃ(企業対消費者)の電子取引や、公的機関への電子申請アプリケーションでは、一般利用者が証拠書類を必要に応じて提示できるよう保存しておかなければならないような状況が存在する。バイヤーから送られてきた注文票、インターネット上で買い物をした場合の領収書、税務申告を行った場合の受領書等がこれらの証拠書類に該当する。この応用例１では、利用者が電子的に送付された証拠書類を紙としてプリントアウトし、保存しておく紙によるフォールバックシナリオの一例である。この紙には、

・アプリケーションデータ(注文票、領収書などの情報)
・アプリケーションデータ(の一部)に対する署名
・上述した方法(１)(２)(３)で述べた再入力支援のための記述
等が、ＸＭＬのタグ付きテキストとして印刷されている。

【００５６】

証拠確認の必要が生じた場合、利用者は保存しておいた紙またはそのコピーを提出する。紙の提出を受けた機関(クレジット会社、税務署など)は、紙から本実施の形態を用いてＸＭＬテキストを再入力し、その内容に基づいて署名を検証する。再入力作業は証拠書類を保存していた利用者、入力を専門に行うサービスプロバイダが行うことも可能である。

【００５７】

図１２は、応用例１におけるＸＭＬデータの例を示した図であり、図の斜体の部分が誤りの防止・検出・訂正に関する情報である。図１２に示すように、ここでは、書籍の注文情報として、明確ではない「－(マイナス)」を置き換えて示している。また、署名情報については、最後にまとめて、誤り訂正情報を記述している。ここでは、バイヤーである「日本太郎」と、署名情報である「Xy6%Dgdeu256&fdi」や「op6&se%$h78s1Wq*ae」に対して、誤り訂正コードが生成されている。

【００５８】

この応用例１のように、本実施の形態によれば、電子的なオリジナルテキストと同一の署名対象データを再現することができる。空白の数や同形文字など、一旦、紙に印刷されてしまうと解り難い(しかし署名の同一性判定には影響する)情報も正確に再入力が可能である。一般に、再入力データに対する署名の検証が失敗した場合、本当にデータに改変が加えられているのか再入力の際に混入した誤りに拠るものなのかを判断し、再入力の際に混入した誤りである場合には、誤りの場所を見付けて修正する、という作業を人手で行う必要がある。本実施の形態を適用すれば、このような手間と時間を要する作業を大幅に簡略化することができる。

【００５９】

また、電子取引や電子申請などのアプリケーションの成否は、小規模な企業や個人がどれだけ参加してくれるかに負うところが大きい。彼らはＷｅｂブラウザを使って取引や申請を行っても、電子的な伝票や証拠をきちんと処理・管理するシステムを通常、備えておらず、運用コストも負担できない。しかしながら、伝票や証拠の類が紙として出力され、電子的表現に容易に戻せることが本実施の形態により保証されていれば、小規模利用者は自身の書類の処理や保管を従来どおり紙ベースで行うことができる。企業間取引においても電子化された形で、発行された注文票が小規模サプライヤにはＦＡＸで届く、といったケースがしばしばあるが、本実施の形態を用いれば、そのような伝票にも容易に証拠能力を持たせて検証することが可能になる。

【００６０】

応用例　２）電子化ワークフローの一部を代替

電子化ワークフローは、企業間/企業内の情報の流れを円滑にし、事務コスト削減やターンアラウンドタイムの短縮などのメリットをもたらす。しかしながら、ワークフロー中のどれか一つの企業/部門が電子化に対応していない場合には、後続の組織はデータの再入力を行うか、そこから先の全てを紙ベースで処理しなければならない。複数の独立性の強い組織(部門や企業)が関連するワークフローでは、各組織のプロセスの電子化レベルが異なっているため、電子化されたワークフローと紙ベースのワークフローとが混在してしまうことが多い。各組織はシステムの開発や更新を個々に実施しており、電子化への重点の置き方も異なっているからである。複数の組織からのトランザクションを一括して処理しなければならない組織にとって、そのトランザクションの電子化は重要であるが、起票元の個々の組織にとってはそれほどの分量にはなっておらず、電子化のプライオリティが低いかもしれないのである。

【００６１】

この応用例２では、例えば、紙ベースの帳票しか受け付けない企業/部門Ｂの前段に位置する企業/部門Ａは、自身が電子的に処理した帳票データを紙として印刷し、後段の企業/部門Ｂに送付する。この紙には、

・帳票データ

・必要なら帳票データ(の一部)に対する署名
・上述の方法(１)(２)(３)で述べた再入力支援のための記述
が、ＸＭＬのタグ付きテキストとして印刷されている。ＸＭＬで記述された帳票データをより人間が見やすい形(例えば表形式)にレンダリングしたものを添付してもよい。

【００６２】

紙帳票を受け取った企業/部門Ｂは、記載されている情報に基づいて処理を行った後、その結果を更に後段の企業/部門Ｃに送付する。このとき、企業/部門Ｂは、企業/部門Ｂが作成した帳票(企業/部門Ｂが修正/追加した情報を含む)に加えて、企業/部門Ａから受け取った紙帳票のコピーを企業/部門Ｃに渡す。紙帳票を受け取った企業/部門Ｃは、人手でまたはＯＣＲを援用して、企業/部門Ａの紙帳票の情報を再入力する。その際、本実施の形態における機能を用いて、入力/認識誤りの自動検出と修正を行うことができる。企業/部門Ｂが作成した帳票の情報の入力については、本実施の形態による支援は望めないが、入力すべき情報量は、企業/部門Ａからの帳票と比べて少ない(企業/部門Ａはそれまで関係した企業/部門が付加/修正した情報の集約)ため、入力側の負担は小さいと予想される。企業/部門Ｃ以降、帳票データは再び電子化されたワークフローによって流通し処理される。

【００６３】

図１３は、この応用例２におけるＸＭＬデータの例を示した図であり、図の斜体の部分が誤りの防止・検出・訂正に関する情報である。ここでは、「交通費」と「書籍」の項目について、「３５００」と「５５００」の料金が記述され、これらの料金に該当する文字列に対して誤り訂正コードが計算されている。このような誤り訂正コードを用いることで、紙からテキストを再入力するときに生じる誤りを自動検出することができ、以後の業務処理等に大切な情報に対する誤りを低減することが可能となる。

【００６４】

応用例 ３）文書の入力支援

例えば、印刷された紙の形でのみ配布された文書(ＸＭＬテキストとは限らな

い)の一部または全体に対し、ときには電子化して利用したいという要求がある。最近の市販ＯＣＲでは、スキャン解像度等の条件が整えば印刷文書をある程度の精度(９５－９９％以上)で読み取ることができ、一次入力手段としては十分に利用可能である。このＯＣＲの出力結果を人手で修正するとき、しばしば問題になるのが文脈処理が効かない専門用語や固有名詞の存在である。これらの語は、認識精度が低くかつ一文書中に特定の語が頻繁に出現するため、修正する側の負担が大きい。専門雑誌、マニュアル、仕様書等にはこういった単語が含まれていることが多い。

【００６５】

この応用例３では、入力担当者は、事前に対象文書を通読するか部分的にＯＣＲ処理することにより、上記のような専門用語や固有名詞を同定し、前述の方法(３)を用いて記述しておく。前の二つの応用例とは異なり、これらの記述はテキストエディタ等で電子的に作成されているものとする。ＯＣＲの中間結果とこれらの記述を組合わせて処理することにより、チェックや訂正に手間のかかる専門用語/固有名詞に対する誤りの自動検出や修正を容易に行うことができる。この応用例３では、入力の対象としてＸＭＬのタグ付きテキストと一般のタグ無しテキストのどちらも扱うことが可能である。

【００６６】

図１４は、この応用例３におけるＸＭＬデータの例を示した図であり、図の斜体の部分が誤りの防止・検出・訂正に関する情報である。ここでは、固有名詞である「鈴木一郎」、「ロゼッタネット」、また、英語の略語である「ＰＩＰ」に対して、訂正情報が付加されている。

【００６７】

応用例 ４）文字化けへの対処

本実施の形態では、紙からの再入力に限らず、データの伝送に関してシステムレベル(トランスポート層)での誤り訂正機能がサポートされていない場合に、その上位レベルである文書交換層やアプリケーション層で誤り訂正を行う一般的な手法として有効である。この応用例４における文字化けへの対処はその一例である。

【００６８】

この応用例４では、ＸＭＬデータ作成者は、文字化けを避けたいテキストに対して、前述の方法(１)(２)を適用して、文字化けの検出/訂正のための情報を付加して作成し、電子的な手段により他者に伝達する。ＸＭＬデータは、複数の媒介者(システムや人)を経て、そのＸＭＬデータの利用者に送られる。文字化けし易いと解っている文字(一部の記号)は、送り出す時点で文字化けを起こさない表現に変換される。仮に、中間過程のどこかで文字化けが起こっていても、誤り訂正情報により訂正するかアプリケーションプログラムで処理する前に警告することができる。

【００６９】

以上、詳述したように、本実施の形態によれば、空白の連続や同形文字など見た目からでは誤りやすい表現を予め別の形で表現して伝えることができる。また、紙からテキストを再入力するときに生じる誤りを自動検出または/および自動修正することが可能となる。更には、紙からテキストを再入力するときに正しく入力された部分については、人間によるチェックを省くことができる。また更に、文字化けし易い文字を別の表現で伝えることができると共に、文字化けを自動検出および/または自動修正することが可能となる。これらの効果は、紙からの再入力に関して人手で入力を行う場合、ＯＣＲ等を援用する場合どちらでも期待することができる。

【００７０】

また、電子的なワークフローにおけるデータ交換、蓄積、処理に関して、本実施の形態によって紙を用いた代替シナリオ(フォールバック)を用意し、実践することができる。文書や帳票の電子化が今後のトレンドであることは間違いないが、ワークフローにおける全ての局面で電子化が行われていないと成立しないようなアプリケーションシナリオでは、参加できる企業/部門は限定されてしまう。本実施の形態のごとく適当な代替シナリオが用意されていることが、文書/帳票の電子化を促進する上で大きな意義を持つと考えられる。更に、ＸＭＬデータの交換・蓄積に関し、日本語プロファイルではＵＴＦ-８かＵＴＦ-１６を推奨しているが、実際にはShift JISや日本語EUC(End User Computing)など様々なエンコ

ーディング方式が使われており、方式間の変換テーブルも一意に決まっていないのが現状である。レガシーシステム(既存システム)との連携を始めると、ベンダーごとに異なる実装がある日本語EBCDIC(Extended Binary Coded Decimal Interchange Code)との変換も必要になってくる。本実施の形態のように、「どこかで文字化けが起こる」と想定して文字化けの防止、検出、訂正のためのデータ記述を用意することで、文字化けが起こらないようなデータ交換の規約作りに依らずとも、一定の効果を得ることが可能となる。

【００７１】

【発明の効果】

以上説明したように、本発明によれば、マークアップによるデータ・文章の記述を行う記述用言語において、テキストを再入力する際に混入し易い誤りや文字化けを検出することができる。

【図面の簡単な説明】

【図１】　本実施の形態における対象データの置き換え例を示した図である。

【図２】　誤り訂正符号の作成例を示した図である。

【図３】　(ａ),(ｂ)は、要素内のテキストに関して誤り訂正情報を挿入した例を説明するための図である。

【図４】　(ａ)～(ｃ)は、本実施の形態における訂正コード記述用属性を用いた訂正情報の挿入例を示す図である。

【図５】　(ａ),(ｂ)は、アプリケーションデータの記述の後に誤り訂正情報を付加した例を示した図である。

【図６】　(ａ)～(ｃ)は、ＯＣＲの文脈処理モジュールに対して提供する情報の例を示した図である。

【図７】　本実施の形態が適用された誤り訂正支援システムの全体構成を示す説明図である。

【図８】　第１のコンピュータ装置１０側の誤り防止・検出・訂正マークアップ付加モジュール１３における処理を示したフローチャートである。

【図９】　第２のコンピュータ装置２０における誤り検出・訂正モジュール

２３内の処理を示したフローチャートである。

【図１０】　中間結果をＸＭＬベースで記述した例を示す図である。

【図１１】　誤り検出・訂正情報付きテキストの処理の概要を示したフローチャートである。

【図１２】　応用例１におけるＸＭＬデータの例を示した図である。

【図１３】　応用例２におけるＸＭＬデータの例を示した図である。

【図１４】　応用例３におけるＸＭＬデータの例を示した図である。

【符号の説明】

１０…第１のコンピュータ装置、１１…第１アプリケーション、１２…マークアップ付加用プロファイル、１３…誤り防止・検出・訂正マークアップ付加モジュール、２０…第２のコンピュータ装置、２１…第２アプリケーション、２２…マークアップ認識用プロファイル、２３…誤り検出・訂正モジュール、３０…データ伝達部、３１…データ送り出し機構、３２…データ受け取り機構、３３…ネットワーク、３４…プリンタ、３５…スキャナ＆ＯＣＲ、３６…ＦＡＸスキャナ、３７…ＦＡＸプリンタ、４０…ＸＭＬアプリケーションデータ、４１…訂正情報付きアプリケーションデータ、４２…書換えＸＭＬアプリケーションデータ、４３…誤り防止・検出・訂正用記述

【書類名】　　図面

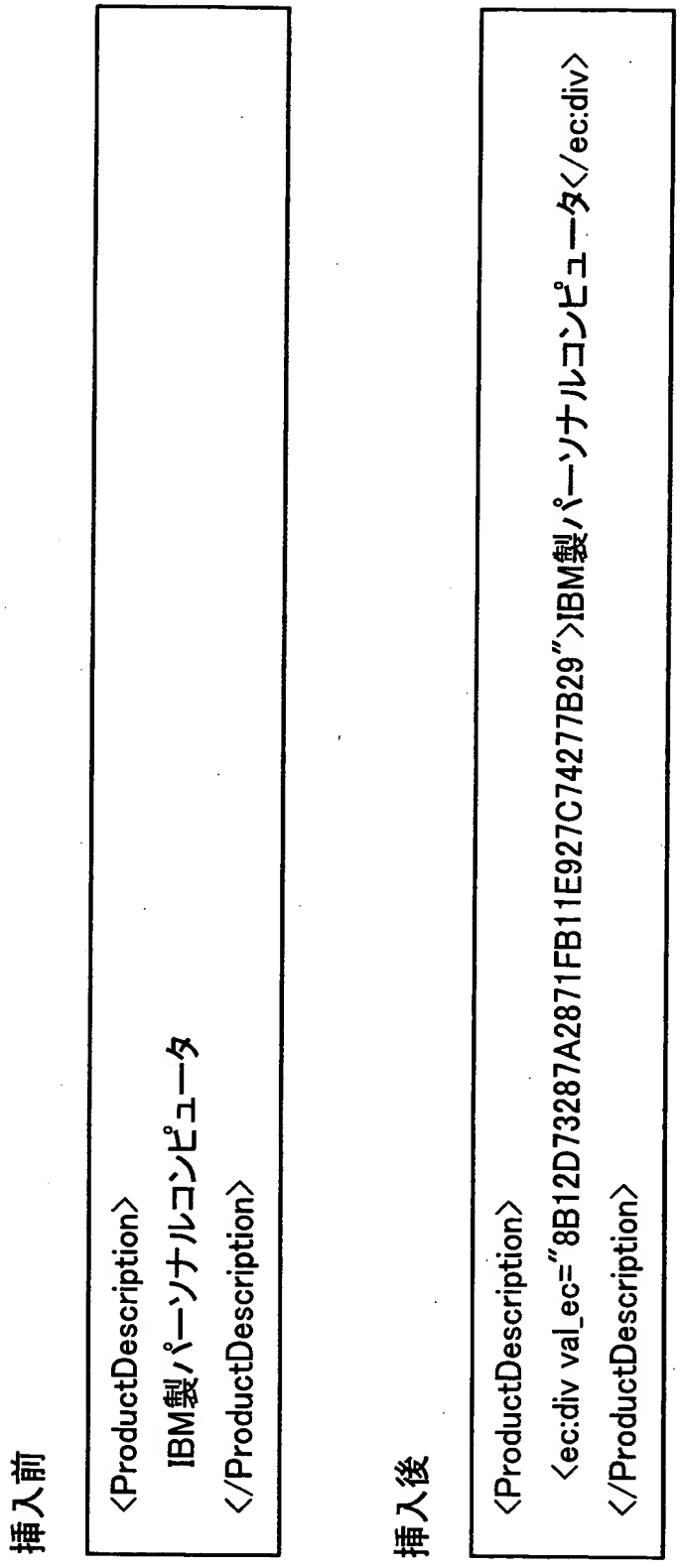
【図１】

置き換え前	置き換え後
（半角空白）	<ec:sp/>
（全角空白）	<ec:sp2/>または<ec:ch utf="x0030"></ec:ch>
－（マイナス）	<ec:ch utf="x0dff">－</ec:ch>
ー（長音）	<ec:ch utf="xfc30">ー</ec:ch>
力（漢字）	<ec:ch utf="x9b52">力</ec:ch>
カ（カタカナ）	<ec:ch utf="xab30">カ</ec:ch>

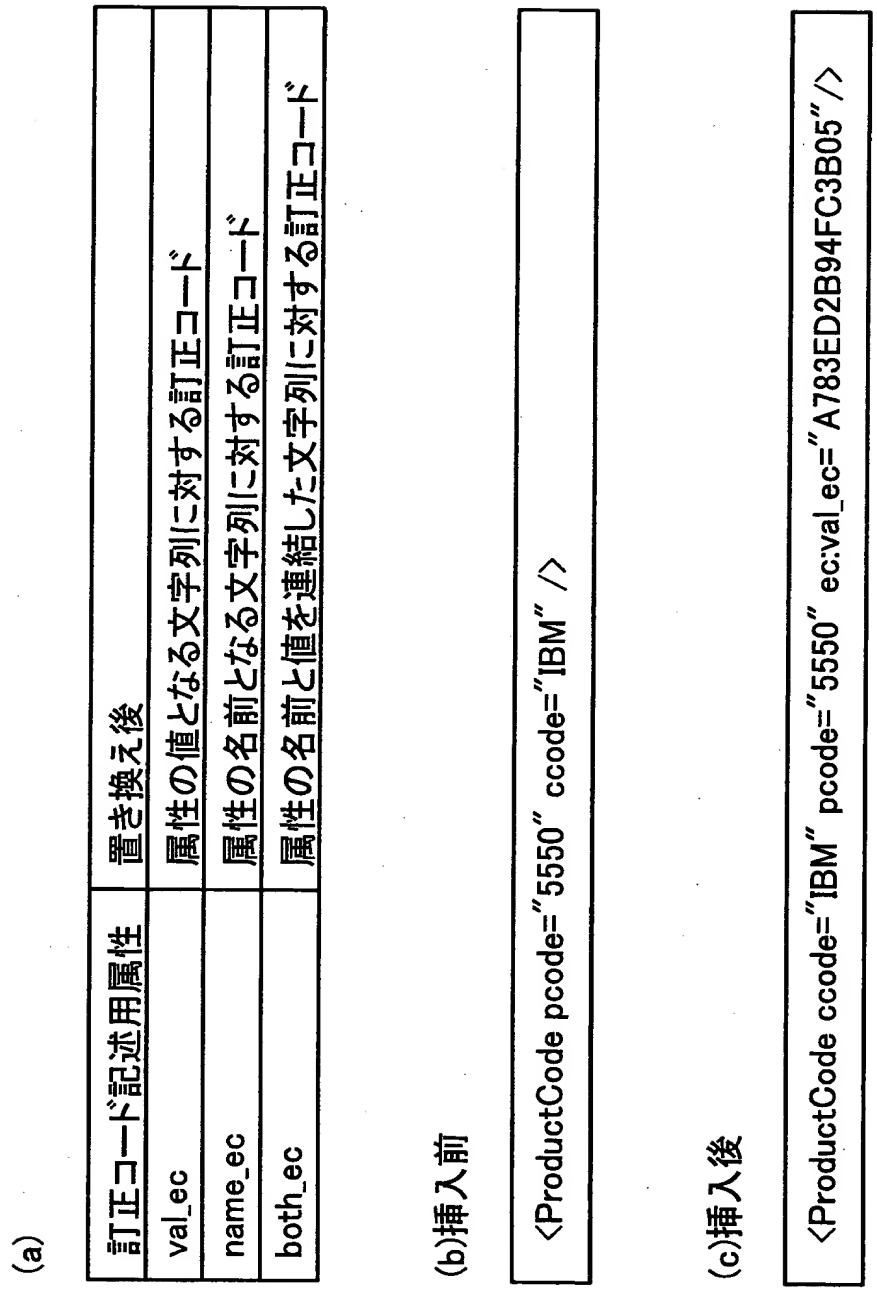
【図２】


コ 1 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0
ン 1 1 1 1 0 0 1 1 0 0 1 1 0 0 0 0
ピ 1 1 0 1 0 1 0 0 0 0 1 1 0 0 0 0
ュ 1 1 1 0 0 1 0 1 0 0 1 1 0 0 0 0
ー 1 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0
タ 1 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0
に 0 1 1 0 1 0 1 1 0 0 1 1 0 0 0 0
よ 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0
る 1 0 0 0 1 0 1 1 0 0 1 1 0 0 0 0
帳 0 0 1 1 0 0 1 1 0 1 0 1 1 1 1 0
票 0 1 1 0 1 0 0 0 0 1 1 1 1 0 0 1
処 1 1 1 0 0 1 1 0 0 1 0 1 0 0 0 1
理 0 0 0 0 0 1 1 0 0 1 1 1 0 1 0 0
は 0 1 1 0 1 1 1 1 0 0 1 1 0 0 0 0
8bitの訂正符号
(eg.2A)を生成
"2A3498CD1B50A81C359EF2A470A08BE6"

【図３】


(a) 挿入前
<ProductDescription>
IBM製パーソナルコンピュータ
</ProductDescription>
(b) 挿入後
<ProductDescription>
<ec:div val_ec="8B12D73287A2871FB11E927C74277B29">IBM製パーソナルコンピュータ</ec:div>
</ProductDescription>

【図４】


(a)
訂正コード記述用属性
置き換え後
val_ec
属性の値となる文字列に対する訂正コード
name_ec
属性の名前となる文字列に対する訂正コード
both_ec
属性の名前と値を連結した文字列に対する訂正コード
(b)挿入前
<ProductCode pcode="5550" ccode="IBM" />
(c)挿入後
<ProductCode ccode="IBM" pcode="5550" ec:val_ec="A783ED2B94FC3B05" />

【図５】

(a)

```
<!-- アプリケーションデータの記述 -->
<ProductDescription>
  IBM製パーソナルコンピュータ
</ProductDescription>
<ProductCode pcode="5550" ccode="IBM" />
<!-- アプリケーションデータに対する誤り訂正情報の記述 -->
<ec:ecc ID="01">
  <ec:div val_ec="7C21E6237893963EC25D23A84952C21E">
    <ec:target ec:path="//ProductDescription/text()"/>
    <ec:target ec:path="//ProductCode/@pcode"/>
    <ec:target ec:path="//ProductCode/@ccode"/>
  </ec:div>
</ec:ecc>
```

(b)

```
<ec:ecc>
  <ec:div val_ec="9E01E6237AB3952F7C21E623D73223AB">
    <ec:target ec:path="//ecc[ID="01"]/div//@*"/>
  </ec:div>
</ec:ecc>
```

【図６】

(a)

```
<ec:word type="ProperNoun">鈴木一郎</ec:word>
<ec:word type="Abbreviation">XML</ec:word>
<ec:word type="TagName">ProductCode</ec:word>
<ec:word type="AttName">ccode</ec:word>
```

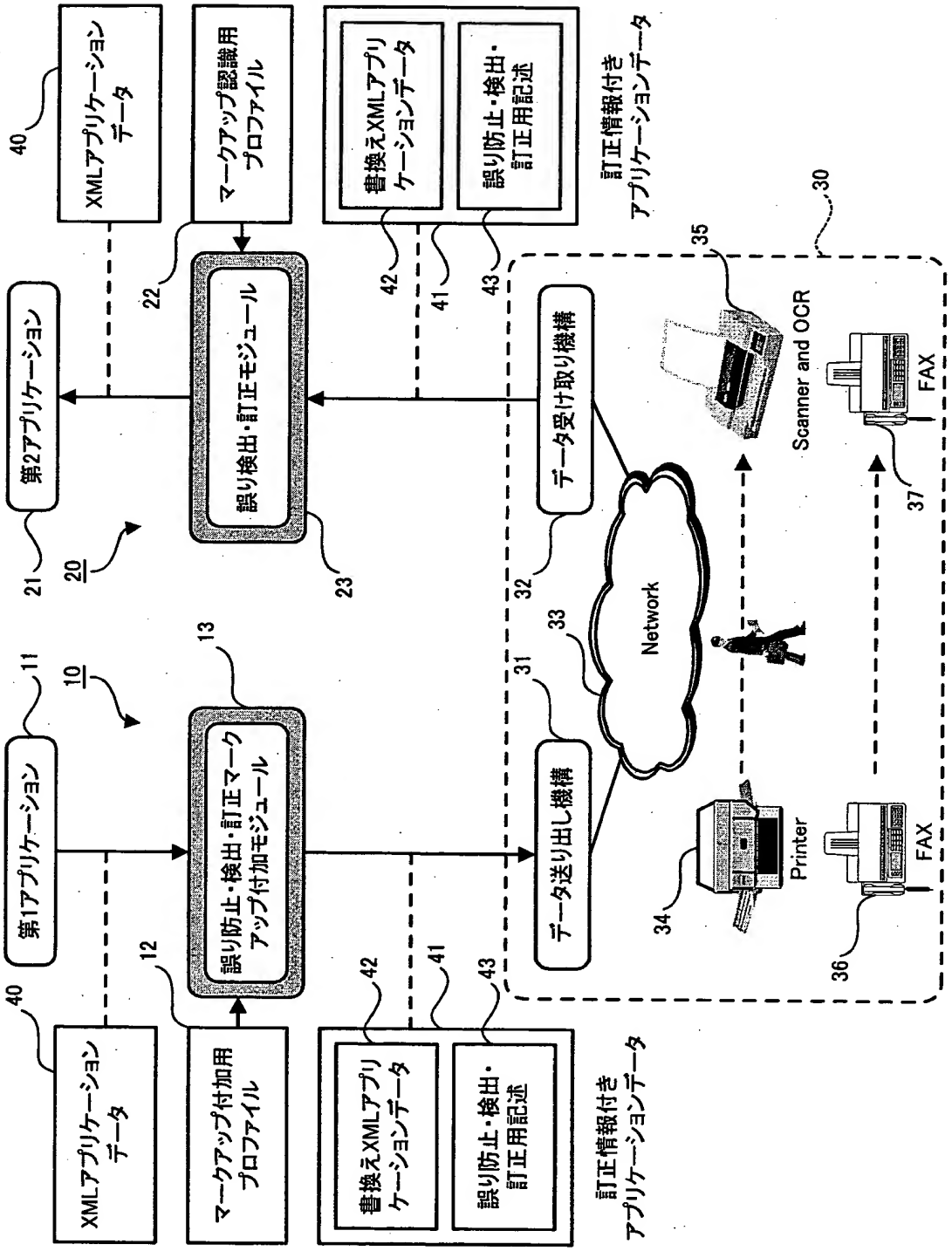
(b)

typeの値	意味
ProperNoun	固有名詞
Abbreviation	英語の略語
TagName	タグの名前
TagVal	要素の値として出現するキーワード
AttName	属性名
AttVal	属性の値として出現するキーワード

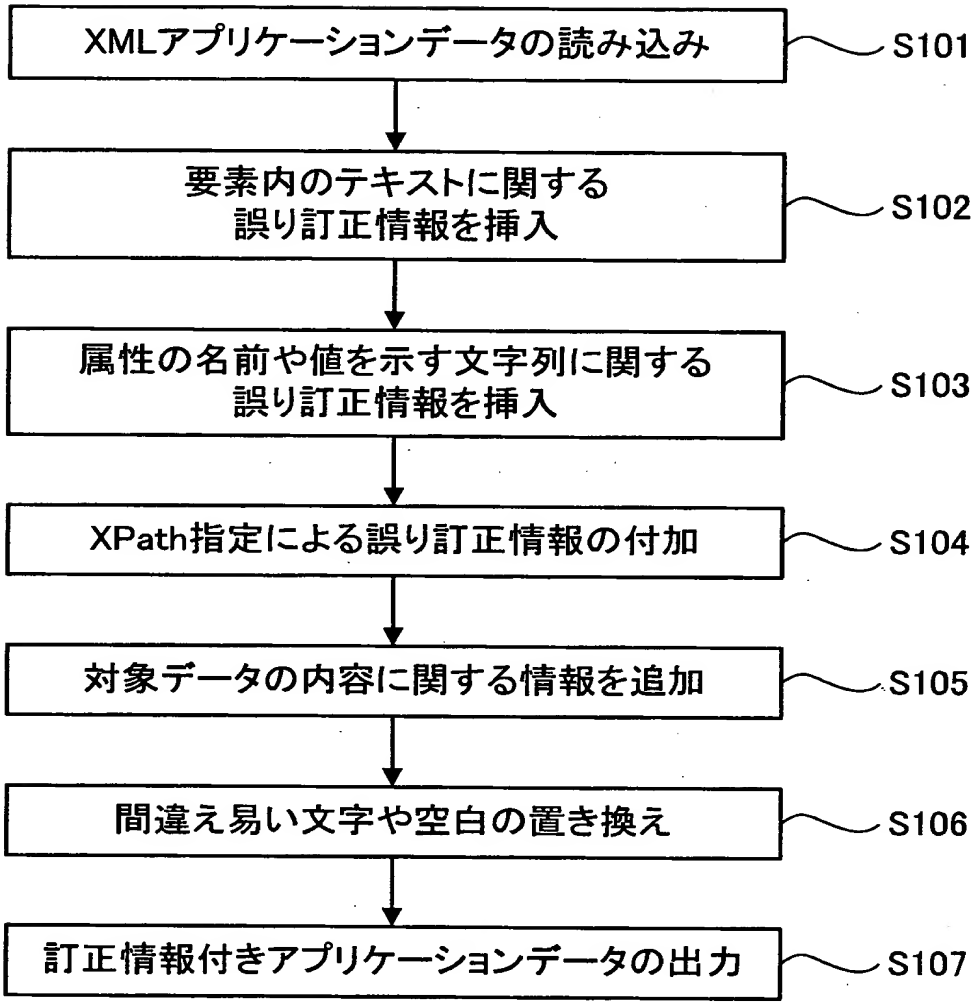
(c)

```
<ec:word type="ProperNoun">
  <ec:div val_ec="7A4D273387C2971FB22E927D74267B2A">
鈴木<ec:ch utf="x004E">一</ec:ch>郎</ec:div>
</ec:word>
```

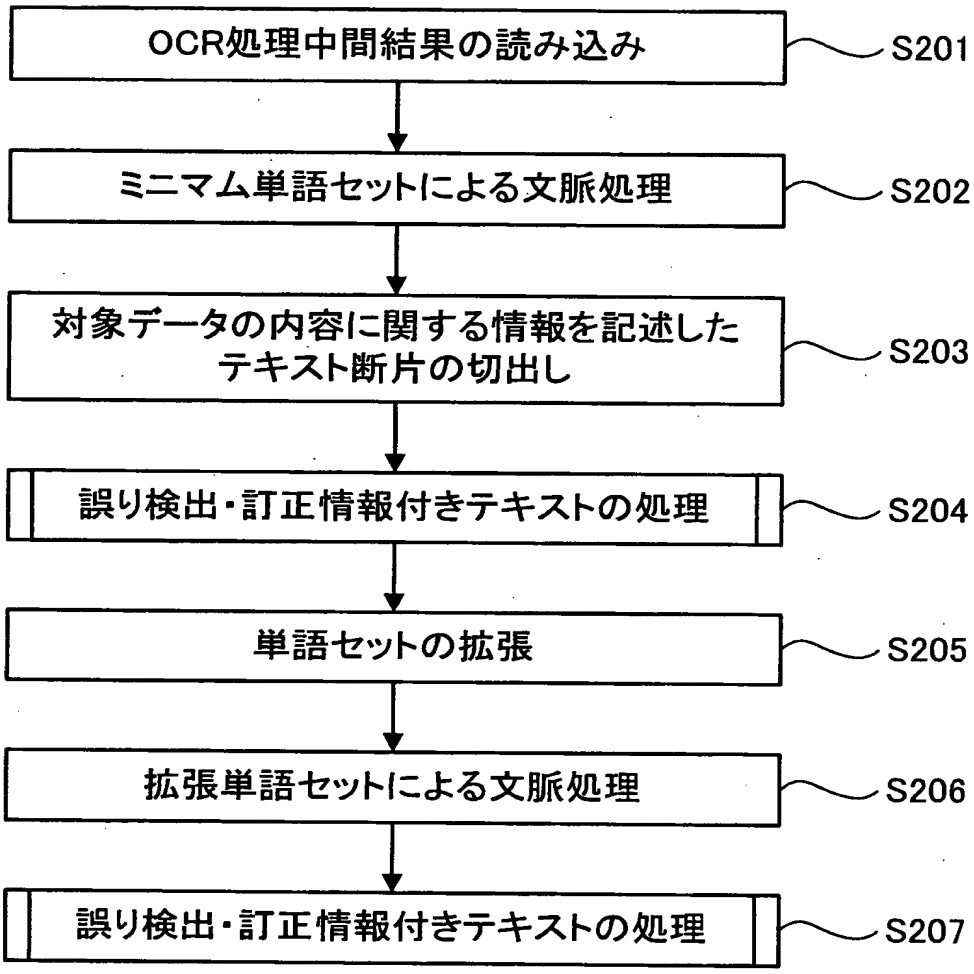
【図７】


XMLアプリケーションデータ
40
マークアップ認識用プロファイル
22
書換えXMLアプリケーションデータ
42
誤り防止・検出・訂正用記述
41
43
訂正情報付きアプリケーションデータ
30
35
第2アプリケーション
21
20
誤り検出・訂正モジュール
23
データ受け取り機構
32
Scanner and OCR
FAX
37
10
第1アプリケーション
11
13
誤り防止・検出・訂正マークアップ付加モジュール
31
33
Network
12
マークアップ付加用プロファイル
データ送り出し機構
Printer
34
FAX
36
訂正情報付きアプリケーションデータ

【図８】

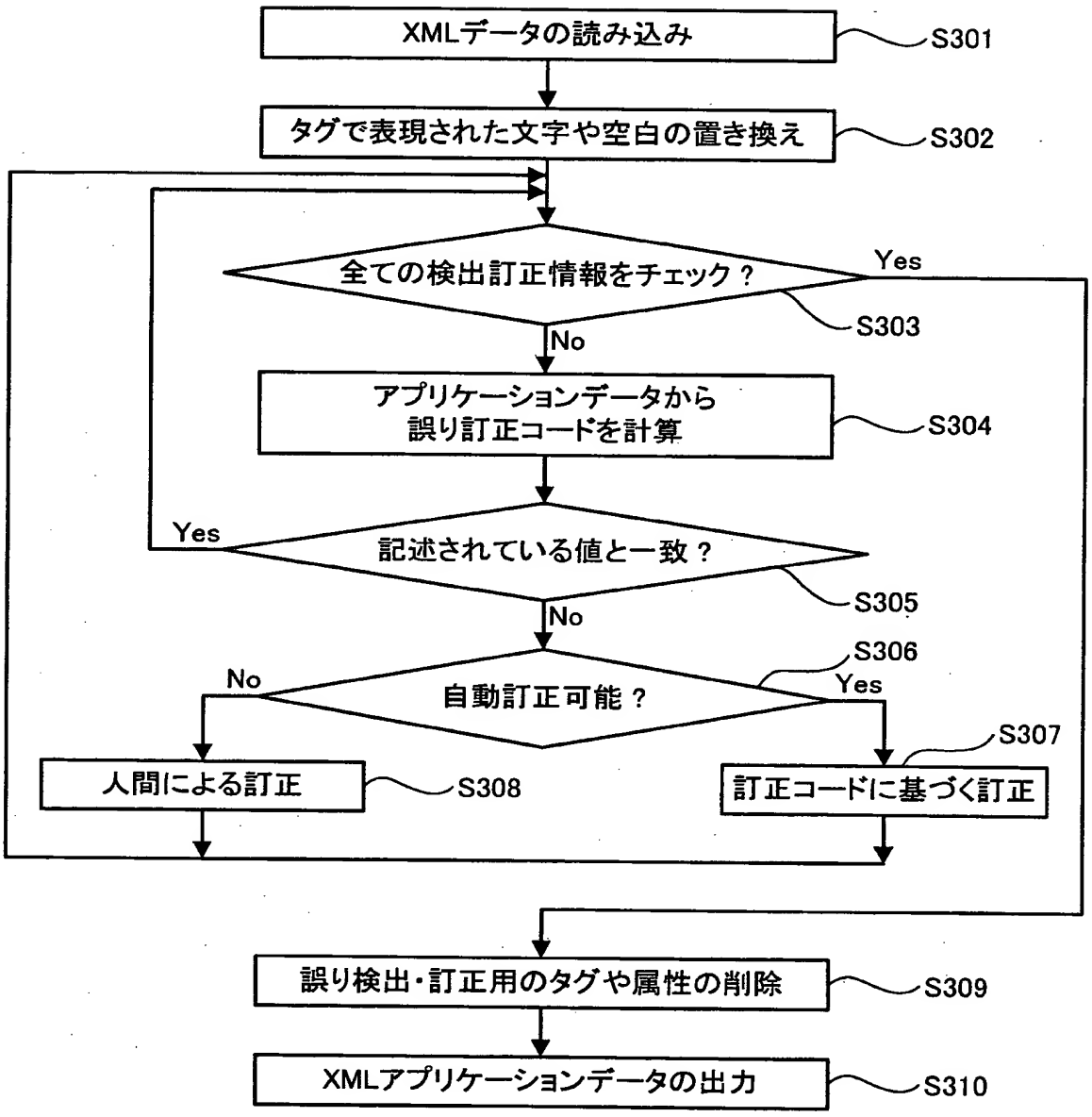

XMLアプリケーションデータの読み込み
S101
要素内のテキストに関する
誤り訂正情報を挿入
S102
属性の名前や値を示す文字列に関する
誤り訂正情報を挿入
S103
XPath指定による誤り訂正情報の付加
S104
対象データの内容に関する情報を追加
S105
間違え易い文字や空白の置き換え
S106
訂正情報付きアプリケーションデータの出力
S107

【図９】


OCR処理中間結果の読み込み
S201
ミニマム単語セットによる文脈処理
S202
対象データの内容に関する情報を記述した
テキスト断片の切出し
S203
誤り検出・訂正情報付きテキストの処理
S204
単語セットの拡張
S205
拡張単語セットによる文脈処理
S206
誤り検出・訂正情報付きテキストの処理
S207

【図１０】

【図１１】


XMLデータの読み込み
S301
タグで表現された文字や空白の置き換え
S302
全ての検出訂正情報をチェック？
Yes
S303
No
アプリケーションデータから
誤り訂正コードを計算
S304
Yes
記述されている値と一致？
S305
No
S306
No
自動訂正可能？
Yes
S307
人間による訂正
S308
訂正コードに基づく訂正
誤り検出・訂正用のタグや属性の削除
S309
XMLアプリケーションデータの出力
S310

【図１２】

```
<PurchaseOrder>
<!-- 書籍の注文情報 -->
<Buyer>
   <Name>日本太郎</Name>
   <CreditCard>9880<ec:ch utf="x0dff">-</ec:ch>2134<ec:ch
utf="x0dff">-</ec:ch>2378</CreditCard>
</Buyer>
<OrderList>
   <Order>
      <ISBN>4<ec:ch utf="x0dff">-</ec:ch>8101<ec:ch utf="x0dff">-
</ec:ch>8979<ec:ch utf="x0dff">-</ec:ch>1</ISBN>
   </Order>
   <Order>
      <ISBN>4<ec:ch utf="x0dff">-</ec:ch>274<ec:ch utf="x0dff">-
</ec:ch>06290<ec:ch utf="x0dff">-</ec:ch>2</ISBN>
   </Order>
</OrderList>

<!-- 署名情報 -->
<Signature>
   <SignedInfo>
      <Transforms>
         <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath
-19991116">
            <XPath ">
               (Buyer | OrderList)
            </XPath>
         </Transform>
      </Transforms>
   </SignedInfo>
   <DigestMethod Algorithm="http://www.w3.org/2000/07/xmldsig#sha1"/>
   <DigestValue>Xy6%Dgdeu256&fdi</DigestValue>
   <SignatureValue>op6&se%$h78s1Wq*ae</SignatureValue>
</Signature>

<!-- 誤り訂正情報の記述 -->
<ec:ecc ID="01">
   <ec:div val_ec="7123E6237893963EC25D23A84952C21E">
      <ec:target ec:path="//Buyer/Name" />
      <ec:target ec:path="//Signautre//text()" />
   </ec:div>
</ec:div>

</PurchaseOrder>
```

【図１３】

```
<ExpenseClaim>
  <Item type="交通費">
    <Date>2000/04/21</Date>
    <Amount><ec:div val_ec="13AB15607A93A63ECF5123C79B51C324">3500</ec:div></Amount>
  </Item>
  <Item type="書籍">
    <Date>2000/05/23</Date>
    <Amount><ec:div val_ec="95A3E5237A93963ECB5023A88C42C21E">5500</ec:div></Amount>
  </Item>
</ExpeseClaim>
```

【図１４】

```
<report>
  <author>鈴木一郎</author>
  <title>データ交換のための新技術</title>
  <chapter>
    <p>本レポートは...</p>
  </chapter>
  <chapter>
    <p>ロゼッタネットはPIPと呼ばれる新しい規格を規定し...</p>
  </chapter>

  <ec:word type="ProperNoun">鈴木一郎</ec:word>
  <ec:word type="ProperNoun">ロゼッタネット</ec:word>
  <ec:word type="Abbreviation">PIP</ec:word>

</report>
```

【書類名】　　要約書

【要約】

【課題】　マークアップによるデータ・文章の記述を行う記述用言語において、テキストを再入力する際に混入し易い誤りや文字化けを検出する。

【解決手段】　第１のコンピュータ装置１０は、ＸＭＬアプリケーションデータ４０における所定部分をタグで置き換えるための情報を含むマークアップ付加用プロファイル１２と、このマークアップ付加用プロファイル１２を参照してアプリケーションデータの所定の部分をタグで置き換えて訂正情報付きアプリケーションデータ４１を生成して出力する誤り防止・検出・訂正マークアップ付加モジュール１３を備え、第２のコンピュータ装置２０は、訂正情報付きアプリケーションデータ４１を入力し、ここに含まれるタグセットを認識してアプリケーションデータ中の誤りや文字化けを検出する誤り検出・訂正モジュール２３を備えたアプリケーションデータ提供システム。

【選択図】　　図７

認定・付加情報

特許出願の番号	特願２０００－２９５００７
受付番号	５００００１２４９１７７
書類名	特許願
担当官	塩崎　博子　　１６０６
作成日	平成１２年１１月１０日

＜認定情報・付加情報＞

【特許出願人】
- 【識別番号】 390009531
- 【住所又は居所】 アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　（番地なし）
- 【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】
- 【識別番号】 100086243
- 【住所又は居所】 神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和事業所内
- 【氏名又は名称】 坂口　博

【代理人】
- 【識別番号】 100091568
- 【住所又は居所】 神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和事業所内
- 【氏名又は名称】 市位　嘉宏

【代理人】
- 【識別番号】 100106699
- 【住所又は居所】 神奈川県大和市下鶴間１６２３番１４　日本アイ・ビー・エム株式会社大和事業所内
- 【氏名又は名称】 渡部　弘道

【復代理人】 申請人
- 【識別番号】 100104880
- 【住所又は居所】 東京都港区赤坂５－４－１１　山口建設第２ビル６Ｆ　セリオ国際特許事務所
- 【氏名又は名称】 古部　次郎

【選任した復代理人】
- 【識別番号】 100100077

次頁有

認定・付加情報（続き）

【住所又は居所】　東京都港区赤坂５－４－１１　山口建設第２ビル
６Ｆ　セリオ国際特許事務所
【氏名又は名称】　大場　充

次頁無

出　願　人　履　歴　情　報

識別番号　　〔３９０００９５３１〕

１．変更年月日　　２０００年　５月１６日
［変更理由］　名称変更
住　所　　アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　（番地なし）
氏　名　　インターナショナル・ビジネス・マシーンズ・コーポレーション